-COMPUTE-

Microprocessor Users Group Newsletter

Volume 5 Number 1 January/February 1979 Price $1.50

TABLE OF CONTENTS

National does not assume any responsibility for use of any circuitry described, no circuit patent licenses are implied, and National reserves the right, at any time without notice, to change said circuitry.

NOTICE TO MEMBERS OF THE USERS GROUP

National Semiconductor sponsors the microprocessor users group and newsletter to provide a forum for the exchange of ideas among users and potential users of National's microprocessors. National maintains the group's software library as a service to members of the users group. National Semiconductor is not responsible for software in the software library, and does not provide applications support for that software. Descriptions published in the newsletter are those of the contributor, and National cannot guarantee their accuracy or applicability. Questions concerning any application or software described in the newsletter should be directed to the person or company whose name accompanies the article. Publication in the newsletter does not constitute endorsement by National Semiconductor.

NATIONAL DEVELOPS 70-SERIES FAMILY OF UNIQUELY CONFIGURED SINGLE-CHIP BUS-ORIENTED NMOS MICROCOMPUTERS

A uniquely configured family of single-chip microcomputer devices with a bus-oriented architecture has been developed by National Semiconductor Corp.

Designated the 70-Series family, the devices are intended for use in systems requiring the economy of single chip, the flexibility of multiprocessing bus architecture, and the power of a comprehensive set of 16-bit arithmetic operations.

Designed for ease of implementation in standalone, direct memory access and multiprocessing applications, first members of the INS8070 family include:

- The INS8070, minimum system containing 64 bytes of RAM and no on-chip ROM
- The INS8072, containing 64 bytes of RAM and 2.5K bytes of ROM

Unlike the input/output oriented single-chip devices now on the market, the INS8070 family is unique in the industry with its bus-oriented architecture. It was developed for single-chip microcomputer users who require a low cost, high capability system solution to their design problems.

The instruction set is especially suited to control system applications, as well as high-level language execution, while the multiprocessing bus architecture makes the design of DMA systems such as terminals, home computers, and word processors very simple.

Communications between INS8070 devices and external memory and/or peripheral devices are effected via a 16-bit dedicated address bus and an 8-bit bidirectional data bus.

Fabricated using National's standard N-channel silicon gate, depletion mode MOS technology, the INS8070 family of

Continued Page 2

devices are each housed in a standard 40-pin dual-in-line package, and require only a single 5-volt supply. Major features of the family include:

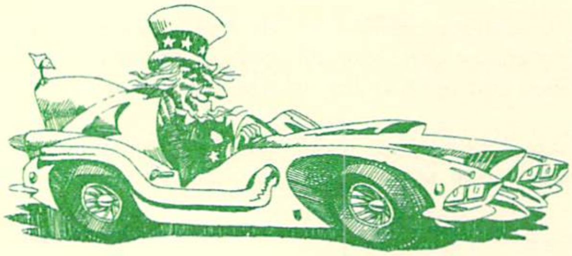
- Simplified programming, with multiple addressing modes, including counter-relative, immediate data, indexed, auto-indexed, implied and single-byte subroutine calls
- Stack operation, with a comprehensive set of 16-bit instructions, including load, store, add, subtract, multiply, divide, exchange, shift, and stack push/pop
- Unique features including an ASCII-to-BCD conversion in one instruction and the character search instruction, both of which are especially useful for high-level language
- Large system capability, with address capability up to 65K bytes of memory
- Handshake bus-access control on-chip, for multiprocessing and DMA operations
- System flexibility, with the ability to interface with memories of peripherals of any speed
- On-chip generation, two interrupt/sense inputs, and three user-accessible control-flag outputs

Separate strobe inputs from the 8070 devices indicate when valid input/output memory or peripheral data are present on the 8-bit bus.

The remaining I/O signals are dedicated to general purpose control and status functions, including initialization, bus management, microprocessor halt, interrupt request, I/O cycle extension, and user-specified hardware/software interface functions.

The INS8070 is also designed to be MICROBUS™ compatible. The MICROBUS concept developed by National defines a universal, component-level system for communication between any microprocessor memory and peripheral devices.

All voltage levels, drive capability, and timing are fully defined. Thus any microprocessor or microprocessor-oriented peripheral that meets these specifications may be interfaced directly, or those that are not entirely compatible may be simply adapted. By the use of this strategy, the MICROBUS concept frees the design engineer from the interfacing chore and allows access to more devices, thus permitting him more design flexibility.



RANDOM DATA

8080 Interrupts - Correction

Many references to the INS8080A, including our technical description booklet, and the Series 8000 Microprocessor Family Handbook, have an error with regard to the 8080A interrupt system operation.

In them, it is stated that when the 8080A recognizes an interrupt it saves both PC and STATUS by pushing them onto the stack. Actually, its response is determined by the particular instruction jammed onto the bus during the interrupt acknowledge cycle.

Only an RST or a CALL instruction will cause the PC to be pushed onto the stack, and in neither case will this save STATUS. If STATUS is to be saved, a PUSH PSW must be executed prior to any other instruction that modifies the program status flags, and a POP PSW should be executed immediately prior to the RET.

# # # # # # # # # #

SM/PL COMPILER

Programmed Logic's version of the SM/PL compiler for the PACE Universal Development System is no longer available. This compiler was described in previous issues of COMPUTE (Vol. 2, No. 8 and Vol. 4, No. 4).

# # # # # # # # # #

MICROCOMPUTER SYSTEMS SERVICE DEPOT

We have changed our return policy for repairs. Please call the Service Information Desk at (408) 737-6270 for a "Service Return Authorization" (SRA number) prior to returning your microcomputer boards or systems for repair. We are located at:

National Semiconductor
675 Almanor Ave.
Sunnyvale, CA 94086
Attention: Service Depot

If you need additional assistance please contact our new manager, Bud Antuna, (408) 737-6598.

Spare parts are available through the Microcomputer Systems Service Information Desk.

The Bit ● Bucket

Dear COMPUTE:

Here are two programs that I have written for the SC/MP: Quad Square Root (SQRT4) and Double Unsigned Multiply (DMPY). As regards the first one, I want you to notice the execution time (which I think is fine) compared to the execution time of the program "SCSQRT" (SL0041A).

The multiplication program is a double-byte version of the "MPY" of the "SC/MP Math Routines" (SL0027A). Notice how I use the CCL instruction, because it can be done the same way in "MPY".

I would be very happy to receive a calculator.

Sincerely,

Jon Kleiser
Kastellveien 9
N-Oslo 11
NORWAY
Telephone 02-688290

Jon's programs (SQRT4 and DMPY) are described on the following pages.

Dear COMPUTE:

I am sending you a copy of a pseudo subroutine (LJSR) that allows the user to do a PC-relative JSR to anywhere in the PACE's address space. This lets the user of LJSR write large position-independent programs. If there is any interest, I have a math library (fixed point) that is position-independent and reentrant.

The cost of using LJSR is that A3 is used by this routine.

The calling sequence is:

```
JSR          LJSR
DATA         FOO-.-1; FOO is the subroutine to call
...          ...
```

Even though LJSR, itself, is position-independent it should be placed in the base page so any program can easily call it.

```
LJSR: PULL A3          ;GET RETURN ADR FROM STACK
      AISZ A3, 1       ;A3 – A3 + 1
      JMP (A1)         ;NO-OP
      PUSH A3          ;PUT UPDATED RTN ADR ON
                       ;STACK
      ADD A3,-1(A3)    ;CALCULATE SUBROUTINE ADR
      JMP (A3)         ;GOTO SUBROUTINE
```

Note: This becomes useful when the displacement of the subroutine is more than ± 127 words away from the JSR.

Sincerely,

Rick Shiffman
Digital Graphics Association
10318 Dunkirk Ave.
Los Angeles, Calif. 90025

Dear COMPUTE:

Here is another copy of my SC/MP Simulator program that I am submitting to the Users Library. I have also written in NIBL a companion program that performs typical monitor functions. Commands Examine, Alter and Move are self-explanatory; Step and Trace use the Simulator program and print out all the registers after each instruction. I wrote it in NIBL simply because it was much easier to do so. As someone else may find it useful I will send it along when it is completed.

If you have any questions or problems please write or call me at (301) 921-3806.

Sincerely,

Bruce F. Field
Electrical Measurements & Standards Div.
Center for Absolute Physical Quantities
United States Dept. of Commerce
Washington, D. C. 20234

A listing of SC/MP Simulator program is available from COMPUTE.

EDITORIAL

If you have a contribution to COMPUTE, send it in and it will be published as quickly as possible. However, the editors cannot promise that everything sent in will be published immediately. All manuscripts should be typed and carefully proofed. All listings and diagrams should be as clear and easy to read as possible.

Most of COMPUTE is made possible by your donations.

Occasionally we get letters or COMPUTE orders where the address is difficult to read. When we translate these to the envelope labels with your library requests or orders they are returned because the postal service cannot find you. If at all possible please use your mailing label on all correspondence to the user group, print legibly or type your name and address.

Thanks a lot.

QUAD SQUARE ROOT SQRT4

by Jon Kleiser
Oslo, Norway

The program SQRT4 takes the square root of a 32-bit unsigned number in RAD1, 2, 3 and 4. The root, which is to be found in RT1 and RT2, is a 16-bit integer whose square is less than or equal to the radicand. The program also calculates the first fractional bit of the root for cases where a round-up is wanted. Then one shall add the content of the Carry/Link flag to the root after return. However, if the input value is greater than 4 294 901 760 (X'FFFF0000), the root will be X'FFFF with the CY/L set, and a round-up will give X'0000.

Register P2 is used as a stack pointer. P3 is the subroutine calling register.

Execution time is from 16140 to 18586 microcycles, i.e., 32.3 37.2 millisec. for an old SC/MP.

The algorithm used is the one for calculating the square root by hand, based on the following formula:

$$(r_1+r_2+...+r_n)^2 = r_1^2+r_2(2r_1+r_2)+...+r_n(2(r_1+r_2+...+r_{n-1})+r_n)$$

Stack usage:

Disp	Name	Function
0	RAD1	Radicand
1	RAD2	--
2	RAD3	--
3	RAD4	--
4	RT1	Root
5	RT2	--
6	RT3	Root frac.
7	DIF1	Difference
8	DIF2	--
9	DIF3	--
A	ONE1	Ones' pos.
B	ONE2	--
C	ONE3	--
D	CNT	Counter

The program:

```
0000   C400   SQRT4:   LDI   0
0002   CA04            ST    RT1 (2)       ; ZERO ROOT
0004   CA05            ST    RT2 (2)
0006   CA06            ST    RT3 (2)
0008   CA0B            ST    ONE2 (2)
000A   CA0C            ST    ONE3 (2)
000C   01              XAE                 ; ZERO RADICAND EXTENSION
000D   C440            LDI   X'40          ; SET ONES' POSITION
000F   CA0A            ST    ONE1 (2)      ; INDICATOR
0011   C411            LDI   17            ; SET BIT COUNTER
0013   CA0D            ST    CNT (2)
0015   C204   NEW:     LD    RT1 (2)       ; ADD "ONE" TO ROOT
1700   DA0A            OR    ONE1 (2)
0019   CA04            ST    RT1 (2)
001B   C205            LD    RT2 (2)
001D   DA0B            OR    ONE2 (2)
001F   CA05            ST    RT2 (2)
0021   C206            LD    RT3 (2)
0023   DA0C            OR    ONE3 (2)
0025   CA06            ST    RT3 (2)
0027   03              SCL                 ; SET CARRY/LINK FLAG
0028   C202            LD    RAD3 (2)      ; SUBTRACT ROOT FROM
002A   FA06            CAD   RT3 (2)       ; HIGHEST PART OF
002C   CA09            ST    DIF3 (2)      ; RADICAND
002E   C201            LD    RAD2 (2)
0030   FA05            CAD   RT2 (2)
0032   CA08            ST    DIF2 (2)
0034   C200            LD    RAD1 (2)
0036   FA04            CAD   RT1 (2)
0038   CA07            ST    DIF1 (2)
003A   40              LDE                 ; DO A POSSIBLE BORROW FROM
003B   FC00            CAI   0             ; RADICAND EXTENSION
003D   06              CSA                 ; CHECK SIGN OF DIFFERENCE
003E   9428            JP    ZERO          ; JUMP IF DIFFERENCE NEG.
0040   C400            LDI   0             ; ZERO RADICAND EXTENSION
0042   01              XAE
0043   C207            LD    DIF1 (2)      ; MOVE DIFFERENCE TO
0045   CA00            ST    RAD1 (2)      ; HIGHEST PART OF
0047   C208            LD    DIF2 (2)      ; RADICAND
0049   CA01            ST    RAD2 (2)
004B   C209            LD    DIF3 (2)
004D   CA02            ST    RAD3 (2)
004F   02              CCL                 ; CLEAR CARRY/LINK FLAG
0050   C206            LD    RT3 (2)       ; ADD "ONE" TO ROOT AGAIN
0052   F20C            ADD   ONE3 (2)      ; TO COMPLETE "THE
0054   CA06            ST    RT3 (2)       ; DOUBLING OF THE ROOT"
0056   C205            LD    RT2 (2)
0058   F20B            ADD   ONE2 (2)
005A   CA05            ST    RT2 (2)
005C   C204            LD    RT1 (2)
005E   F20A            ADD   ONE1 (2)
0060   CA04            ST    RT1 (2)
0062   9016            JMP   SHIFT
0064   909A   SQRT4X:  JMP   SQRT4         ; LONG DISTANCE JUMP
0066   90AD   NEWX:    JMP   NEW           ; SPLIT-UPS
0068   C204   ZERO:    LD    RT1 (2)       ; REMOVE THE "ONE"
006A   E20A            XOR   ONE1 (2)      ; ADDED PREVIOUSLY
006C   CA04            ST    RT1 (2)
006E   C205            LD    RT2 (2)
0070   E20B            XOR   ONE2 (2)
0072   CA05            ST    RT2 (2)
0074   C206            LD    RT3 (2)
0076   E20C            XOR   ONE3 (2)
0078   CA06            ST    RT3 (2)
007A   C203   SHIFT:   LD    RAD4 (2)      ; SHIFT RADICAND LEFT
007C   F203            ADD   RAD4 (2)      ; (A CCL WAS NOT NECESSARY)
007E   CA03            ST    RAD4 (2)
0080   C202            LD    RAD3 (2)
0082   F202            ADD   RAD3 (2)
0084   CA02            ST    RAD3 (2)
0086   C201            LD    RAD2 (2)
0088   F201            ADD   RAD2 (2)
008A   CA01            ST    RAD2 (2)
008C   C200            LD    RAD1 (2)
008E   F200            ADD   RAD1 (2)
0090   CA00            ST    RAD1 (2)
0092   40              LDE
0093   70              ADE
0094   01              XAE
0095   C20A            LD    ONE1 (2)      ; SHIFT ONES' POSITION
                                           ; INDICATOR RIGHT
0097   1F              RRL                 ; (A CCL WAS NOT NECESSARY)
0098   CA0A            ST    ONE1 (2)
009A   C20B            LD    ONE2 (2)
009C   1F              RRL
009D   CA0B            ST    ONE2 (2)
009F   C20C            LD    ONE3 (2)
00A1   1D              SRL
00A2   CA0C            ST    ONE3 (2)
00A4   BA0D            DLD   CNT (2)       ; DECREMENT BIT COUNTER
00A6   9CBE            JNZ   NEW           ; JUMP IF NOT FINISHED
00A8   C206            LD    RT3 (2)       ; SHIFT FRACTION INTO
00AA   F206            ADD   RT3 (2)       ; CARRY/LINK FLAG
00AC   3F              XPPC  3             ; RETURN
00AD   90B5            JMP   SQRT4X
```

DOUBLE UNSIGNED MULTIPLY DMPY

by Jon Kleiser
Oslo, Norway

The program DMPY multiplies two 16-bit unsigned numbers in MRH, MRL and MNDH, MNDL. The 32-bit product is to be found in PROD1, 2, 3 and 4.

Register P2 is used as a stack pointer. P3 is the subroutine calling register.

Stack usage:

	Entry:		**Return:**	
Disp	**Name**	**Function**	**Name**	**Function**
FE	SHIFT	Shift reg.	SHIFT	
FF	CNT	Counter	CNT	
00	MRH	Multiplicator	PROD1	Product
01	MRL	---	PROD2	---
02	MNDH	Multiplicand	PROD3	---
03	MNDL	---	PROD4	---

The program:

```
0000   C200   DMPY:    LD   MRH (2)      ; GET HIGHER PART OF MULTIPLIER
0002   CAFE            ST   SHIFT (2)    ; SAVE IN SHIFT
0004   C201            LD   MRL (2)      ; GET LOWER PART OF MULTIPLIER
0006    01             XAE               ; SAVE IN E-REG
0007   C400            LDI  0            ; ZERO HIGHEST HALF OF PRODUCT
0009   CA00            ST   PROD1 (2)
000B   CA01            ST   PROD2 (2)
000D   C410            LDI  16           ; SAVE COUNTER
000F   CAFF            ST   CNT (2)
0011    02    $LOOP:  CCL                ; CLEAR CARRY/LINK FLAG
0012    40             LDE               ; LOAD LOWER PART OF MULTIPLIER
0013   D401            ANI  1            ; TEST LEAST BIT
0015   9828            JZ   NO           ; JUMP IF BIT IS ZERO
0017   C201            LD   PROD2 (2)    ; ADD MULTIPLICAND (16 BITS)
0019   F203            ADD  MNDL (2)     ; TO 32-BIT RESULT
001B   CA01            ST   PROD2 (2)
001D   C200            LD   PROD1 (2)
001F   F202            ADD  MNDH (2)
0021    1F    NOADD:   RRL               ; SHIFT
0022   CA00            ST   PROD1 (2)
0024   C201            LD   PROD2 (2)
0026    1F             RRL               ; SHIFT
0027   CA01            ST   PROD2 (2)
0029   C2FE            LD   SHIFT (2)
002B    1F             RRL               ; SHIFT. (PRODUCT IS SHIFTED
002C   CAFE            ST   SHIFT (2)    ; INTO SHIFT/E-REG WHILE
002E    40             LDE               ; MULTIPLIER IS SHIFTED OUT)
002F    1D             SRL               ; SHIFT
0030    01             XAE
0031   BAFF            DLD  CNT (2)      ; DECREMENT COUNTER
0033   9CDC            JNZ  $LOOP        ; JUMP IF NOT FINISHED
0035   C2FE            LD   SHIFT (2)    ; GET LOWEST HALF OF PRODUCT
0037   CA02            ST   PROD3 (2)
0039    40             LDE
003A   CA03            ST   PROD4 (2)
003C    3F             XPPC 3            ; RETURN
003D   90C1            JMP  DMPY
003F   C200   NO:      LD   PROD1 (2)
0041   90DE            JMP  NOADD
```

SC/MP and SC/MP-II Users Group

SC/MP users may be interested in the existence of a SC/MP and SC/MP-II Users Group which has been formed. Members may take advantage of a library of both software and hardware information available on a cost basis. In addition, a bibliography of SC/MP articles, advertising, programming hints, etc., is available to members for the reproduction cost. The construction of a homebrew system based on the SC/MP-II is also in the planning.

No dues or fees are involved. However, in order to receive the monthly newsletter, send a self-addressed stamped envelope to Tom Bohon, 2215-A Walker Dr., Omaha, NB 68123.

PASCAL BIBLIOGRAPHY

National soon hopes to announce the addition of the PASCAL high-level language to our systems. Meanwhile, we have compiled the following bibliography of books and articles pertaining to PASCAL.

Bowles, K. L., MICROCOMPUTER PROBLEM SOLVING USING PASCAL,
Springer-Verlag, New York, 1977.

Bowles, K. L., INTRODUCTION TO COMPUTER SCIENCE,
Springer-Verlag, New York, 1977.

Conway, R., Gries, D., and Zimmerman, C., A PRIMER OF PASCAL,
Winthrop Publishers, Boston, 1976.

Conway, R., Gries, D., and Zimmerman, C., A PRIMER ON STRUCTURED PROGRAMMING USING PASCAL,
Winthrop Publishers, Boston, 1976.

Grogono, Peter, PROGRAMMING IN PASCAL,
Addison-Wesley, Reading, Mass., 1978.

Jensen, K., and Wirth, N., PASCAL - USERS MANUAL AND REPORT,
Springer-Verlag, New York, 1974.

Nori, K. V., Ammann, U., Jensen, K., Nageli, H. H., and Jacobi, Ch., THE PASCAL (P) COMPILER: IMPLEMENTATION NOTES, 1974.

Schneider, C. Michael, Weingart, Steven, and Perlman, David M., INTRODUCTION TO PROBLEM SOLVING AND PROGRAMMING WITH PASCAL,
Wiley, New York, 1977.

Shillington, K., Ander, S. D., Fanning, S., and Bennett, L., UCSD (Mini-Microcomputer) PASCAL, Release Version September 1978, I.5, Institute for Information Systems, La Jolla, Ca., 1978.

Wirth, Niklaus, ALGORITHMS + DATA STRUCTURES = PROGRAMS, Prentice-Hall, Englewood Cliffs, N.J., 1976.

Wirth, Niklaus, SYSTEMATIC PROGRAMMING - AN INTRODUCTION, Prentice-Hall, Englewood Cliffs, N.J., 1973.

BYTE Magazine, August 1978.

The PASCAL NEWSLETTER is published quarterly by the PASCAL User's Group for $6.00 per year. Contact Andy Mickel, University of Minnesota Computer Center, 227 EX, 208 SE Union Street, University of Minnesota, Minneapolis, MN 55455

SC/MP BOOTSTRAP LOADER

by Robert P. Haviland
2100 S. Nova Road, Box 45
Daytona Beach, Florida 32019

The short bootstrap loader program shown below is for beginning SC/MP users who work with Baudot tape or equivalent. It should also be of interest to others who need such a loader.

Assumptions:

- Baudot tape, including start bit, five data bits, two stop bits; start is bus bit 0.
- Serial (SI, SO) input and output, SENSE B = Character Ready.
- Two successive Baudot characters per eight-bit word.
- Baudot bit 5 indicates a control character (only rubout used).
- Baudot bits 1-4 entered in binary coded hex (Baudot A = Hex 3, etc.)
- Single step used to set starting address, to escape loader, and to jump to program execute.
- Register use as shown in program.
- For ASCII, can omit combining operations, and add checksum and more control characters.

NOTES:

1. To execute, single step to 11, then enter +51 (Hex 33). This jumps to 3E.
2. To escape, single step to 11, then enter +53 (Hex 35) and proceed, or use reset.

```
Address   Data (Hex.)   Mnemonic    Comment

  00         08         NOP         Single Step
  01         C4         LDI
  02         --         --          Manual enter Hi Start Address
  03         36         XPAH-P2
  04         C4         LDI
  05         --         --          Manual enter Lo Start Address
  06         32         XPAL-P2
  07         06         CSA         Auto Step: Wait Loop: Check Status
  08         D4         ANI
  09         20         20
  0A         98         JZ-PC       Skip if character started
  0B         FB         -5
  0C         8F         DLY         Delay 11 ms
  0D         0D         13          (for 2.5 MHz clock)
  0E         19         SIO         Get a bit
  0F         06         CSA         Recheck Status
  10         D4         ANI
  11         20         20
  12         98•        JZ-PC       Skip if end of character
  13         04         +4
  14         8F         DLY         To End of Bit
  15         0D         13
  16         90         JMP-PC      Skip if another bit
  17         F4         -12
  18         19         SIO         End of character. add Stop bit
  19         C4         LDI         Check control character
  1A         FE         255
  1B         60         XRE
  1C         98         JZ-PC       Skip if not control
  1D         E9         -23
  1E         06         CSA         Check which half
  1F         D4         ANI
  20         01         01
  21         9C         JNZ-PC      Skip if second half
  22         0E         +14
  23         C4         LDI         Set a mask
  24         1E         31
  25         50         ANE         Get 1/2 byte
  26         1E         RR          Shift
  27         1E         RR
  28         1E         RR
  29         1E         RR
  2A         CA         ST-PZ       Store Temporarily
  2B         00         0
  2C         C4         LDI         Set a flag
  2D         01         01
  2E         07         CAS
  2F         90         JMP-PC      Go to Wait Loop
  30         D6         -42
  31         C4         LDI         Set a mask
  32         1E         31
  33         50         ANE         Get 1/2 byte
  34         DA         OR-P2       Combine
  35         00         00
  36         1E         RR          Normalize
  37         CE         ST@P2       Store, increment
  38         01         +1
  39         C4         LDI         Clear flag
  3A         00         00
  3B         07         CAS
  3C         90         JMP         Go to wait
  3D         C9         -55
  3E         92         JMP-P2      End of load -2
  3F         FE         -2          Then to Start of Load
```

DISK-ORIENTED DEVELOPMENT SYSTEM, WITH INTEGRATED HARDWARE AND SOFTWARE, SPEEDS MICROCOMPUTER SYSTEM DESIGN

A new interactive, microcomputer development system, with fully integrated hardware/software and high-level languages, dramatically cuts microprocessor system hardware and software development time. Called the STARPLEX™ Development System by National Semiconductor Corporation, the system is designed specifically to provide the functions needed by the user in a systems-development environment. Integrated hardware and software provides the key to the improved human engineering aspects of this development system.

Priced at $13,800, STARPLEX costs significantly less and has much more capability than other development systems. STARPLEX incorporates three microprocessors, 64K bytes of memory, dual 256K byte floppy disks, a 1920 character video monitor, a standard keyboard, a function keyboard and a 50 cps printer in a modular enclosure suitable for tabletop use. Also included is a sophisticated operating system complete with text editor, macroassembler, loader, linker, BASIC and FORTRAN software.

STARPLEX is based on a National BLC 80/204 central processor, BLC 8229 CRT/keyboard controller and a BLC 8221 floppy disk controller, each with an 8-bit INS8080A microprocessor. Use of three microprocessors permits the peripheral controllers to relieve the central processing unit of many overhead tasks. Both CRT and disk controllers are capable of direct memory access.

Random-access memory is a 64K word BLC 8064 memory board while standard bulk storage is a dual, floppy-disk drive with a total formatted storage of 512K bytes.

Integrated into the 25 by 16 by 26 inch enclosure is a quiet 50-character per second thermal printer, a 12-inch, 80-character by 24-line green phosphor CRT display and a 96-key multifunction keyboard. The central keyboard is a standard 58 key ASCII unit with upper/lower case capability. Three other key-pads provide CRT control and special operating functions. The system has four open chassis slots for system options and system expansion with standard Series/80 I/O cards. STARPLEX options include an in-system emulator (ISE), PROM programmer, and standard Series/80 I/O expansion cards.

The STARPLEX operating system has a consistent command set and a common file and input/output structure, regardless of the language used. Programmers who write in FORTRAN can access files written in BASIC or assembly language and vice versa. Memory and peripherals are treated as files. Transfers among them are handled automatically, the three internal microprocessors supplying the necessary protocol.

Editing, normally the most time-consuming task, has been given specific attention. Depressing the EDIT key initiates the STARPLEX editor, displaying a menu into which the programmer enters the file name and other editor control. Editing functions are accomplished exclusively with a single key stroke, with the system managing the mass storage file. Using the cursor-control key pad, the designer quickly selects the appropriate character or line. He can insert characters or lines, hold a line in place and roll the frame to reposition it or page through the file.

Assembler, linker and file programs include similar prompting features. If during any operation, the programmer becomes disoriented, a HELP key provides a more descriptive narration, similar to that which would be found in an operator's manual. ERROR messages are displayed in plain English. An audible tone signals an illegal operation, and the system will disregard entry.

The STARPLEX in-system emulator is a sophisticated peripheral that aids software development and speeds integration of microprocessor system hardware and software.

Because of its sophistication, compact nature, and price, STARPLEX can also function as a small computer in industrial instrumentation, communications and small-business applications. The flexibility, price and performance of the STARPLEX system place it at the confluence of microprocessor/microcomputer development systems, intelligent terminals and small minicomputer systems.

For information on STARPLEX, contact your nearest National sales office, local franchised distributor, or call National Semiconductor toll-free (800) 538-1866 outside California; (800) 672-1811 in California.

III III III III III III III

BLC 80/10 COMPATIBLE 8-CHANNEL COMMUNICATIONS BOARD

SYSCOM of Santa Clara, California, recently announced the availability of a new 8-channel serial communications board for users of BLC 80/10 microcomputers. The board features eight independently controlled USARTs with software selectable Baud rates to 9600 Baud, and a choice of asynchronous or synchronous transmission protocols in full-duplex configurations. A hardware timer and jumper selectable interrupts are also provided. The I/O interfaces are all EIA RS-232C compatible with optional on-board modules available for eight optically-isolated high-level current loop interfaces (15-100mA, 12V to 150V). Board size is 6.75" x 12" x 0.6".

For more information and ordering procedures, contact:

Bala Parasuraman
SYSCOM, Inc.
2996 Scott Blvd.
Santa Clara, Calif. 95050
(408) 246-2437

USER'S LIBRARY ORDER FORM

Program		Check for Program Listings	Source Paper Tapes		
Number	Name		Number of Tapes	Cost Each	Total Cost
PACE PROGRAMS					
SL0015A	PACRAM			$ 5.00	$
SL0018A	CALCULATOR		NA		
SL0022A	NUMPRG			$ 5.00	$
SL0025A	PALM			$ 5.00	$
SL0026A	TABTAP			$ 5.00	$
SL0029A	BINBCD		NA		
SL0032A	DIVIDE		NA		
SL0033A	DELSEM			$ 5.00	$
SL0035A	PRNTLM			$ 5.00	$
SL0036A	BASCII			$ 5.00	$
SL0037A	JITTER			$ 5.00	$
SL0048A	BIORHYTHM			$ 5.00	$
SL0054A	8080-X			$25.00	$
SL0059A	PACEFLPT			$15.00	$
SL0061	8048-X			$25.00	$
SC/MP PROGRAMS					
SL0027B	SC/MP MATH			$ 5.00	$
SL0039A	TAPE I/O			$ 5.00	$
SL0041A	SCSQRT			$ 5.00	$
SL0043A	NIBL-N		*	$15.00	$
SL0043A	NIBL-P		*	$15.00	$
SL0049A	REFLEX			$ 5.00	$
SL0047A	PLOT			$ 5.00	$
SC/MP NIBL PROGRAMS					
SL0051A	DISASSEMBLER			$ 5.00	$
SL0060	NIBASM		NA		
SC/MP MACRO PKG					
SL0052A	MDSMAL				
SL0053A	MDSCMPL			$ 5.00	$
PDP-15 PROGRAMS					
SL0045A	PACE-X			$ 5.00	$
PDP-8 PROGRAMS					
SL0046A	SC/MP-X		NA		
PDP-11 PROGRAMS					
SL0057	PACE-X			$15.00	$
SL0058	SC/MP-X			$15.00	$

*Price includes manual, program listing, and paper tape load module

NOTES:

1. Please make sure the programs you select are for the microprocessor you have.
2. There is no charge for program listings, but the number of listings per order is limited to three.
3. NA indicates not available.
4. The prices quoted are in DLR for Australia and in dollars for the United States and Germany.

NAME ____________________

TITLE ____________________

COMPANY ____________________

ADDRESS ____________________

CITY ____________________

COUNTRY ____________________

Fill out the form completely, make your check payable to COMPUTE, and mail to:

UNITED STATES
COMPUTE/208
National Semiconductor Corp.
2900 Semiconductor Drive
Santa Clara, CA. 95051

JAPAN
COMPUTE
NS International Inc.
Miyake Bldg. 6F, 1-9 Yotsuya
Shinjuku-Ku, Tokyo 160, Japan

EUROPE
COMPUTE
National Semiconductor GmbH
808 Fuerstenfeldbruck
Industriestrasse 10, West Germany

HONG KONG
COMPUTE
NS Electronics (HK) Ltd.
4 Hing Yip St., 11th Floor
Kwun Tong, Kowloon, Hong Kong

AUSTRALIA
COMPUTE
NS Electronics Pty. Ltd.
Cnr. Stud Rd. & Mtn. Highway
Bayswater, Victoria 3153
Australia